

Dartmouth College

## Dartmouth Digital Commons

---

Dartmouth College Undergraduate Theses

Theses and Dissertations

---

6-1-2003

# An Evaluation of the Impact of Models for Radio Propagation on the Simulation of 802.11b Wireless Networks

Evan W. Richardson  
*Dartmouth College*

Follow this and additional works at: [https://digitalcommons.dartmouth.edu/senior\\_theses](https://digitalcommons.dartmouth.edu/senior_theses)



Part of the [Computer Sciences Commons](#)

---

### Recommended Citation

Richardson, Evan W., "An Evaluation of the Impact of Models for Radio Propagation on the Simulation of 802.11b Wireless Networks" (2003). *Dartmouth College Undergraduate Theses*. 29.  
[https://digitalcommons.dartmouth.edu/senior\\_theses/29](https://digitalcommons.dartmouth.edu/senior_theses/29)

This Thesis (Undergraduate) is brought to you for free and open access by the Theses and Dissertations at Dartmouth Digital Commons. It has been accepted for inclusion in Dartmouth College Undergraduate Theses by an authorized administrator of Dartmouth Digital Commons. For more information, please contact [dartmouthdigitalcommons@groups.dartmouth.edu](mailto:dartmouthdigitalcommons@groups.dartmouth.edu).

# **An Evaluation of the Impact of Models for Radio Propagation on the Simulation of 802.11b Wireless Networks**

Evan Richardson

June 1<sup>st</sup>, 2003

Senior Honors Thesis

Adviser: Luiz Felipe Perrone

## **Abstract**

Working with an existing wireless network simulator, we describe the addition of both a method for modeling arbitrary terrain, and for calculating signal attenuation with the Irregular Terrain Model (ITM). We also investigate ITM's effects on upper protocol layer in comparison to the Two-Ray Ground Reflection model. Upon examination, it was found that aside from the terrain between the transmitter and receiver, ITM's various parameters are of little significance in the computed signal attenuation. Further, examination of the behavior of the upper protocol layers revealed that at high traffic levels, choice of propagation model can have significant effects on the results of the simulation.

# 1 Introduction

The widespread use of wireless networking has increased the importance of communication protocols capable of decreasing user lag times and making the best use of available radio bandwidth. However, due to the large size and high complexity of a network with more than a non-trivial number of nodes, it is very difficult to analytically assess and optimize the behavior of such a system. It has therefore become important to create realistic simulations of wireless networks in order to better understand their behavior.

There has been moderate understanding of the effects of the simulated radio frequency (RF) propagation model on the upper protocol layer performance of the network as a whole. In many cases, only the most general, least realistic models of RF propagation involving a flat terrain are used in order to reduce computing time while providing what is often thought to be an acceptable approximation of moderately flat terrain. The accuracy of this assumption is not well known. While the GloMoSim team, working with their own wireless network simulator, has shown that the modeling of RF propagation can change some of the simulation results significantly [2], the importance of terrain modeling in RF propagation for nearly flat terrain remains questionable.

In order to better understand the simulation of the RF propagation layer, we implemented terrain modeling, and the Irregular Terrain Model (ITM) for calculating signal attenuation. These sub-models were added to the wireless simulator developed at Dartmouth's ISTS: the Simulator for Wireless Ad-hoc Networks (SWAN). The most important contributions of our work, however, were an evaluation of ITM's sensitivity to various parameters and an experimental investigation of the effects of detail in RF propagation models on the simulation of higher-level protocols.

The remainder of this paper is structured as follows. In Section 2, we briefly describe the workings of SWAN and its subsystems. Section 3 deals with the terrain sub-model and its implementation, while Section 4 describes ITM in depth. Section 5 describes our sensitivity analysis of ITM, and Section 6 describes the effects of the simulated terrain on the simulation as a whole. Finally, we draw conclusions on our work in Section 7, while Section 8 describes possibilities for future work.

## 2 Simulation Framework

### 2.1 DaSSF: Dartmouth Scalable Simulation Framework

SSF is an interface created by a consortium of research institutions for the construction of scalable simulation models [3]. SSF allows the modeler to concentrate on the construction of the simulation *model* and not have to dabble in the details of the simulation *engine*. The structure of the SSF API and its object-oriented nature allow models to be more easily partitioned for parallel or distributed execution.

DaSSF is the implementation of the SSF API developed at Dartmouth College and claims high-performance computing as its primary goal. DaSSF provided the simulation engine

upon which SWAN was built; it allows for the simplified modeling of various components of the simulation infrastructure.

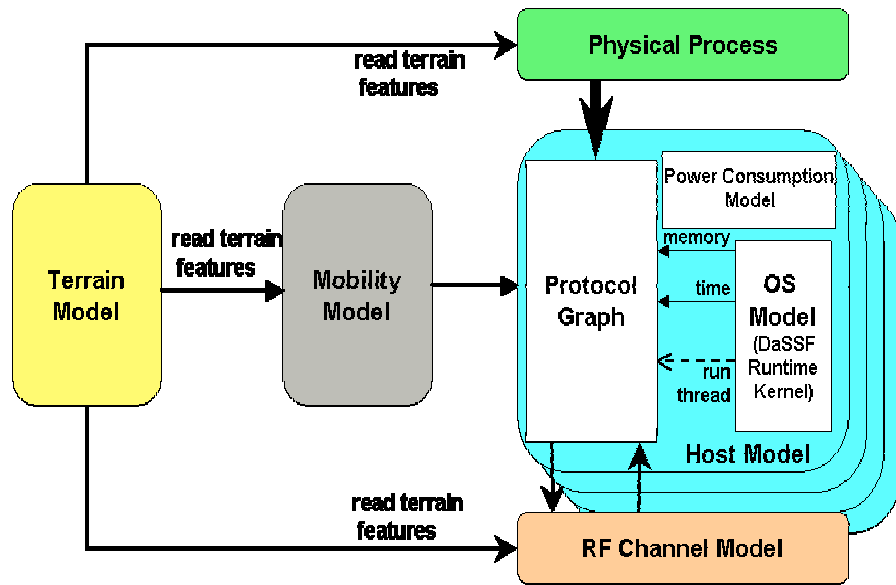
DaSSF implements methods to read files in a format known as Domain Modeling Language (DML). DML is a model description language with simple but powerful syntax. The language can be described as a recursively-defined list of key and attribute pairs, as is shown Figure 2.1. The use of DML allows greater freedom and efficiency in simulation since recompilation need not take place every time some parameter of the simulation is altered. Instead, any of a number of precompiled sub-models may be replaced in the DML to alter the performance of the simulator for any of the available sub-models. This capability is used extensively in SWAN and in our implementation of ITM. Further information about DaSSF and DML, along with the DaSSF source code can be found at [4].

```
DML ::= attribute-list
attribute-list ::= empty | attribute-list attribute
attribute ::= key value | key [ attribute-list ]
key ::= [a-zA-Z_][a-zA-Z0-9_]*
value ::= INTEGER | FLOAT | STRING
```

**Figure 2.1:** The DML grammar [6]

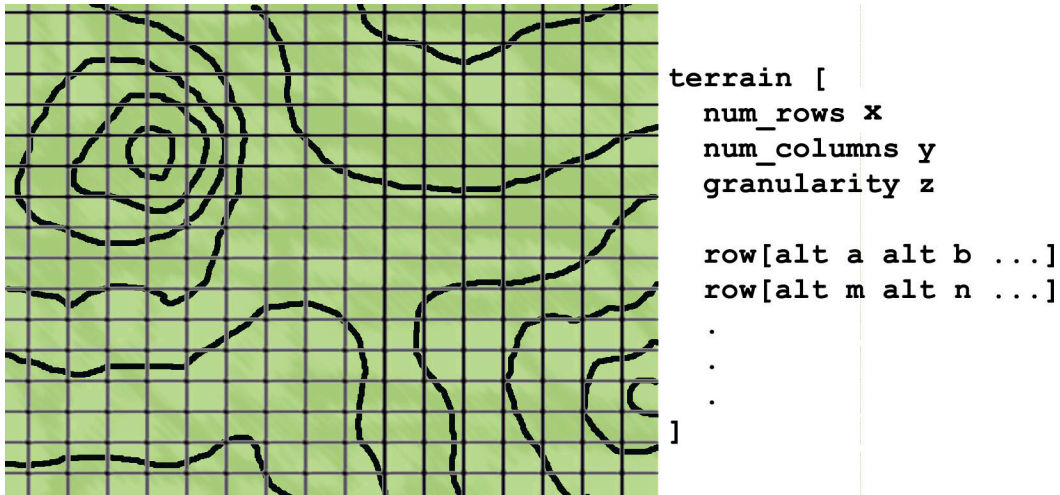
## 2.2 SWAN: Simulator for Wireless Ad-Hoc Networks

SWAN is simulator for wireless networks. It provides for the simulation of all layers of the protocol stack, along with the simulation of the environment in which the nodes are placed. Built on top of DaSSF in order to provide highest performance, SWAN allows for the simulation of a large number of nodes, either mobile or stationary, in arenas of variable size running any type of application the user desires to create. SWAN uses DML files to define the models and their parameters.



**Figure 2.2:** An overview of SWAN's architecture

As seen in Figure 2.2, SWAN is made up of five major components. The most basic of these is the Terrain sub-model; it has the capability to create correlations between the Physical Process, Mobility and RF channel sub-models. Prior to our implementation, the topology of the space that was modeled was always flat. One of our contributions was to define a base class for the description of arbitrary terrain models. From this base class we derived a description of a digital elevation map in the form of an arbitrarily fine altitude grid, as is seen in Figure 2.3, to describe the terrain. Our implementation provides the capability to specify the altitude of points on a rectangular grid and export this data to other terrain-conscious sub-models.



**Figure 2.3:** An example of a topographical map with a grid superimposed. Each of the grid's intersection points corresponds to a data point in our altitude grid. To the right is an example of a DML terrain file

The second sub-model we implemented was a version of ITM that can be used as an

alternative to the 2-ray ground reflection (TGR) and Friis Free Space (FFS) models of RF propagation used in SWAN. ITM requires a profile of the simulated terrain to compute the path loss between the two points. This profile is provided by the terrain sub-model (see Section 3). ITM adds a considerable amount of realism to a portion of the simulation that before preferred computational performance to realistic simulation of the radio channel. The channel model takes information from the Terrain sub-model, if required, and uses it along with other parameters provided in the DML file to compute a gain for simulated signals between antennas. ITM is described in depth in Section 4.

Physical processes can also be affected by the terrain, and reflect events that take place in the outside world. At present, SWAN provides a simple model for the diffusion of gasses over a defined area, but other obvious examples include the simulation of weather conditions or animal movements. Any physical process that would interact with nodes in the network would be simulated by this sub-model. As the development of SWAN continues, this sub-model will be central to any comprehensive simulation of real-world events.

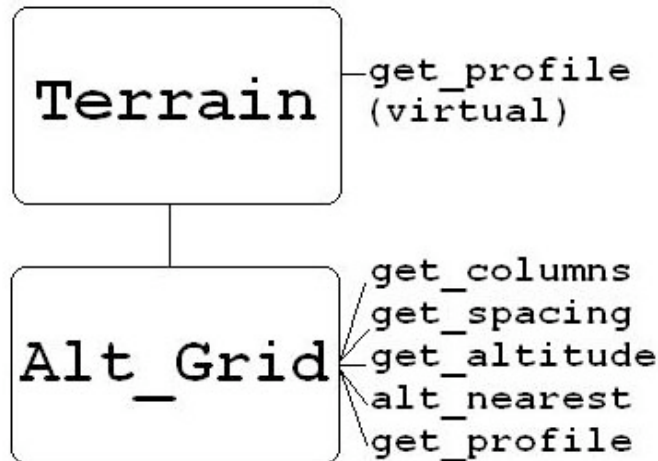
Various different mobility models (stationary, random waypoint) have been implemented in SWAN. These models determine the movement of nodes within the simulated arena. Further realism will be added to these models of motion when terrain data is incorporated into their velocity and direction. Clearly, the mobility of a node and its movement throughout the network have significant effects on the behavior and performance of the network. In this study, which assesses the accuracy of radio propagation models, we have considered only the stationary model.

The last main sub-model of SWAN gives each modeled node a simulated protocol stack. This sub-model provides for the simulation of everything from the physical layer to the application layer. Depending on the modeler's needs, various simulated implementations can be used at each layer. Many of the most common, such as the 802.11b PHY and MAC layers, IP, ARP, AODV, DSR and various models of traffic sources, are provided in the current implementation of SWAN.

### **3 The Terrain Sub-Model for SWAN**

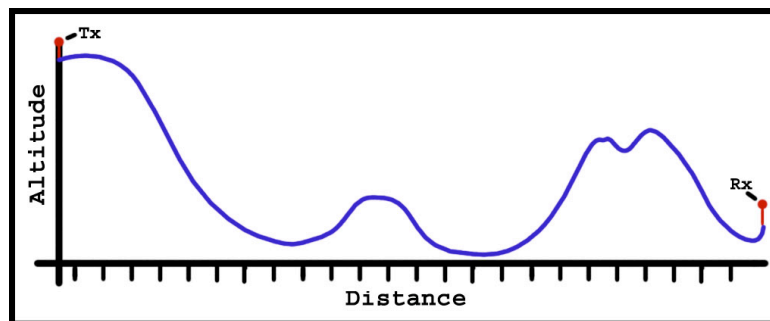
In our implementation of the terrain sub-model, we began with the creation of a terrain base class similar in nature to the Mobility, Physical Process and RF channel base classes. This class serves as an abstract data type definition for more specific terrain classes, and gives a general specification for the creation of future terrain sub-models. Within the base class, we added a virtual function for the creation of a profile.

Derived from the Terrain class is the Alt\_Grid class. This class was written specifically for use with ITM, though it could be used by the environmental or mobility sub-models in the future. It stores the altitude grid and associated data, as shown in Figure 3.1 and implements a method for deriving a terrain profile from the grid along with start and end points.



**Figure 3.1:** The class hierarchy of the terrain sub-model

The terrain profile can be thought of as a thin slice of the terrain that separates the antennas of the transmitter and receiver, as shown in Figure 3.2. It serves to provide ITM with altitude information for the ground between the two nodes. With this information, ITM calculates a path loss value that takes account of reflection, scattering and diffraction effects resulting from the terrain between the sender and receiver, and the given combination of environment parameters.

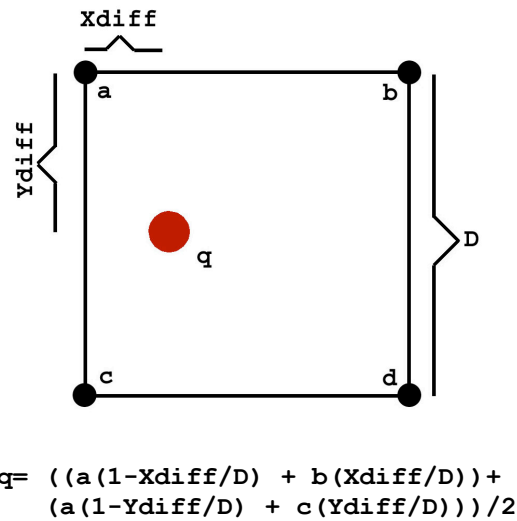


**Figure 3.2:** An example of a terrain profile, starting at the sender, ending at the receiver.

The USGS version of ITM (see Section 4) requires that a large number of points be provided in the terrain profile in order for accurate results to be computed. We followed the USGS MS Windows version's lead in requiring that all profiles contain 800 points between the transmitter and receiver. While it might seem more logical to require constant separation between points, and provide ITM with a terrain profile whose number of points varied with each calculation, this would have opened accuracy and memory allocation issues for very short and very long distances, respectively. While these could have been avoided by minimum and maximum limits on the number of points in a

profile, such arbitrary limitations seemed little better than the 800 point fixed size imposed by the USGS. Further, a constant size for all profiles allowed for more efficient compile time allocation of a static profile array to be re-used for each call of ITM.

Since virtually all points will not fall directly on a data point of the altitude grid, we needed to make use of interpolation when creating our profile. The interpolation routine we designed consisted of taking the weighted average of the 3 nearest points, as shown in Figure 3.3, where D is the distance between points in the altitude grid. The value is weighted based on the proximity of the interpolated point to the 3 reference points. The weighted average in the X and Y dimensions is found, and these two values are averaged together. This is a simple linear interpolation algorithm in two dimensions, and it thus assumes that the change in terrain is constant between recorded points. Such an algorithm tends to provide better accuracy than an algorithm that simply takes the altitude of the nearest recorded point as the interpolated value because it provides smooth interpolated terrain instead of stair-step interpolation values.



**Figure 3.3:** The interpolation algorithm.

## 4 The Irregular Terrain Model

ITM is a computational model that estimates the signal attenuation over a given path from a set of variables in either a point-to-point or area modes. The area mode provides techniques to estimate the path-specific parameters so that gain can be computed when no terrain map is available [10]. ITM's point-to-point mode, used exclusively in our implementation because of our access to terrain data, allows for the calculation of gain between two antennas separated by a well-defined terrain in the form of a terrain profile.

The ITM point-to-point mode takes a set of parameter values and a profile of the terrain between the antennas of the transmitter and receiver, and returns a predicted gain. These parameters, excluding the profile, along with their units, ranges are listed in Table 4.1. They provide the user with the ability to precisely define the environment that he wishes to simulate, and allow for great diversity in the range of environments that ITM can



model.





ITM was developed from the Longley-Rice model for computation of terrain-conscious signal attenuation [10]. Our decision to use ITM in our modeling of a terrain-sensitive network simulation was based largely on its versatility. Its dependence on a simulator-created profile allows for arbitrary accuracy in the modeling of terrain while letting the programmer independently determine the method for modeling the simulated profile. Further, ITM's use of additional variables, ranging from the general climate type to the conductivity of ground, allows the user to determine to a very high level of specificity the type of environment which he desires to simulate.

Parameter	Units	Range	Description
Confidence	% points	(0, 100)	The level of confidence that the loss returned will be greater than the actual value.
Reliability	% points	(0, 100)	
Radio Climate	Arbitrary	[Equatorial, Desert]	An arbitrary definition of the climate type in which transmission occurs
Antenna Height 1	(m)	Greater than Zero	The height of the transmitting antenna
Antenna Height 2	(m)	Greater than Zero	The height of the receiving antenna
Polarity	Arbitrary	[0, 1]	Whether the wave is vertically or horizontally polarized
Dielectric Constant		[4, 25]	
Ground Conductivity	A complex, dimensionless number	[0.001, 0.02]	Surface transfer impedance of the ground
Carrier Frequency	Hz	Greater than Zero	The frequency of the carrier wave
Minimum Surface Refraction	N-units	[320, 310]	Minimum monthly mean surface refractivity

**Figure 4.1:** Parameters of the ITM radio propagation model

Our implementation of ITM is based on code provided by the United States Geological Survey [5]. The USGS, under Fred Najmy and Alaka Paul have created a MS Windows based implementation of ITM that performs both area and point-to-point predictions of gain. This modern version is based on an older Fortran implementation of ITM; the MS Windows version merely provides a Fortran to C++ translation with a GUI. While the entire code body of the USGS project is not available, the code for the point-to-point mode has been made public. This code was used after making the changes that allowed ITM to function in SWAN. In validating our version we checked its results against those of the USGS's MS Windows version using the same variable settings and terrain profiles

(Figure 4.2). We found the two implementations maintained good precision, with the two versions maintaining an average difference of less than .5 db over various terrain profiles, as shown in Figure 4.2. While exact equality between the two implementations was not achieved, the small difference likely arose as a result of unknown code differences between MS Windows and SWAN.

Profile approximation	USGS (dB)	SWAN ITM (dB)
	133.8	132.5
	132.4	132.388
	133.7	133.2
	131.2	131.2
Av. Diff.		-0.453

**Figure 4.2:** Path losses computed by two different implementations of ITM for a set of four different terrain profiles.

## 5 Sensitivity Analysis of ITM

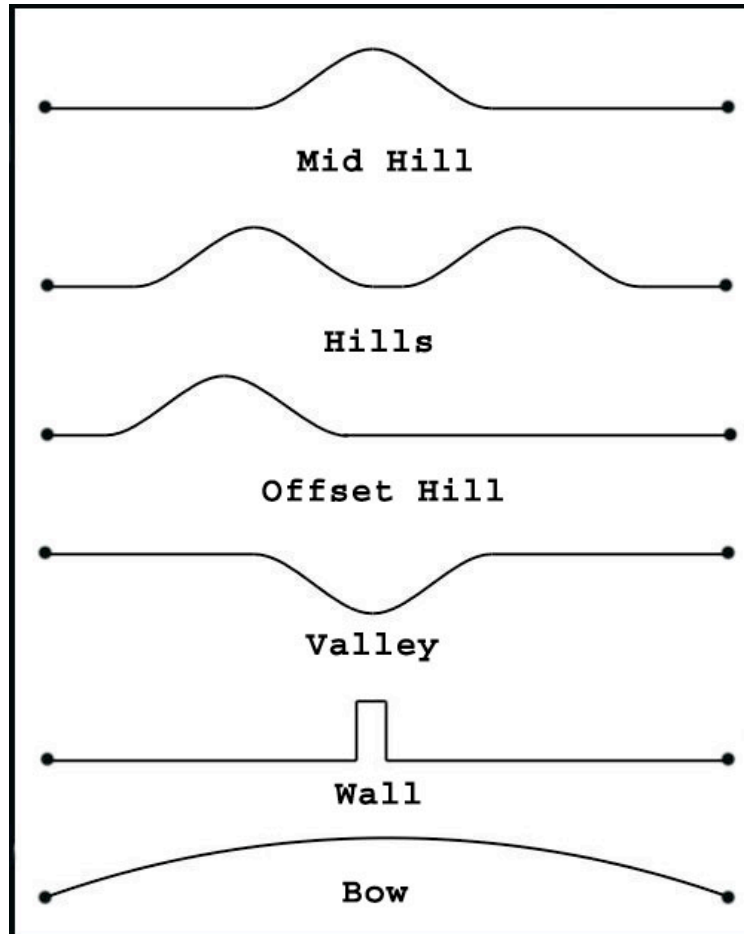
In order to gain a better understanding of the behavior of ITM, we ran several experiments investigating its sensitivity to changes in model parameters. Our goal was to assess the magnitude of each parameter's effects, and the form these effects took on the computed signal attenuation.

In order to carry out these experiments efficiently, we made a few changes in the SWAN/ITM code base. We effectively eliminated the terrain and its computation of the necessary profile, and substituted an additional DML file that contained one of the profiles given in Figure 5.1. This allowed us to examine the importance of each of ITM's variables while utilizing specific profiles.

For this set of experiments, we utilized a system called Scripts for Organizing 'Spiriments (SOS) [12], which automates the experimentation process by pre-defining the interesting variables and their ranges for each experimental set. SOS automates the execution of the various experiments and places their results into a MySQL database. SOS also contains scripts to automate the extraction of this information for use with GNUPlot. SOS has been modified to work with SWAN and is an integral component of the SWAN distribution.

In our experiments with ITM, we decided to use a high and low value for each variable and, together with several terrain profiles, find the gain for every possible permutation of the parameters. For the parameters, excluding the profile, we took the high and low values suggested by the USGS in their MS Windows implementation of ITM. This choice of values seemed obvious because the USGS had been intimately involved in the design of ITM, and would have therefore had a good understanding of its limitations and real

ranges that environmental variables might experience. While the real world user will not likely use values at the extreme edge of the acceptable range, we found these values to be most appropriate for our purposes because they were likely to show the widest variance and thus give us the best picture of how and to what degree each variable affected the gain computed by ITM. Figure 5.2 shows the range of values assigned to each of the variables.



**Figure 5.1:** Different terrain profiles used in our experiment. Note that these figures are not drawn to scale; they serve only to determine the shape of the terrain profiles.

Since ITM calculates signal attenuation based in large part on the reflection, scattering and diffraction effects that result from the terrain between the two antennas, it was important to test ITM's performance over a variety of terrain profiles. In creating those profiles used in our experimentation with ITM, we chose a set of shapes that would approximate a large percentage of the profiles that would be found in the real world. This set can be seen in Figure 5.1. We decided that the maximum amplitude for any landform would be 5 meters in order to maintain both experimental uniformity and the expectation that some reasonable amount of the transmitted wave front would arrive at the receiving antenna. Hereafter, we will use the term 'landform' to describe the terrain in the real world, and 'profile' to describe the digital model of the terrain that we create.

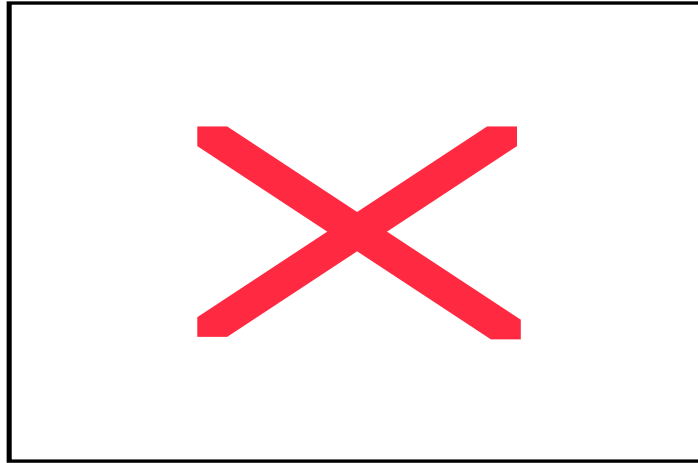
In designing the profiles, we tested the effects of a change in distance on the computed signal attenuation, and so created 3 versions of each landform, one each for 50, 100, and 300 meters of separation. While the separation changed, we maintained the same landforms, and therefore had to change more than the separation between points for each of the profiles. Each landform was created out of half, then a third again as many points for each respective increase in distance. Thus, the landform itself does not change in an absolute sense; only the distance between antennas changes.

Parameter	Values
Confidence	50.0
Reliability	10.0
Radio Climate	[1, 4] (Equatorial, Desert)
Antenna Height 1	1.5
Antenna Height 2	1.5
Polarity	[0, 1]
Dielectric Constant	[4, 25]
Ground Conductivity	[0.001, 0.02]
Carrier Frequency	2.4GHz
Minimum Surface Refraction	[320, 310]

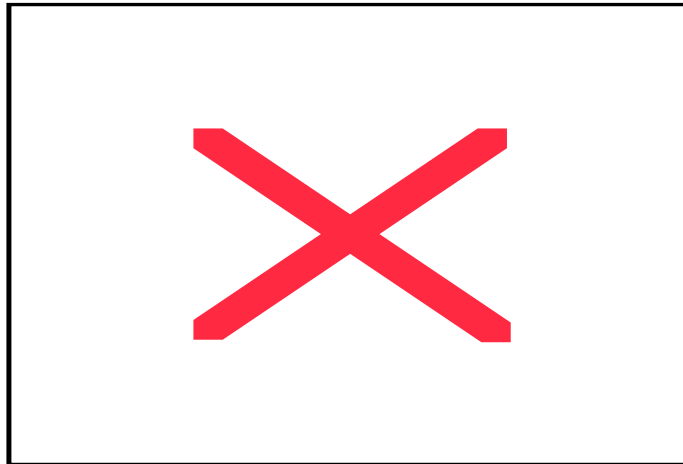
**Figure 5.2:** Parameter values used in the sensitivity analysis of ITM

Our results from this set of experiments with ITM suggest that while the model has a great degree of customizability to fit various environments, this customization matters very little for the 802.11b signal frequency (2.4GHz) and ranges of roughly 350m in open space.

Both the environment and ground conductivity parameters showed negligible effect for the tested values in our experimentation. Results of gain computation were virtually identical for all permutations. Both variables showed approximately 0.001 dB change. The effects of changes to the dielectric constant were also small: roughly 0.02 dB. The largest effects on the computed signal attenuation were observed when we varied Polarity and Minimum Surface Refraction, changed the computed gain by approximately 0.2 dB, as shown in Figure 5.3.



**Figure 5.3:** A representative sample of the dB value changes due to each variable



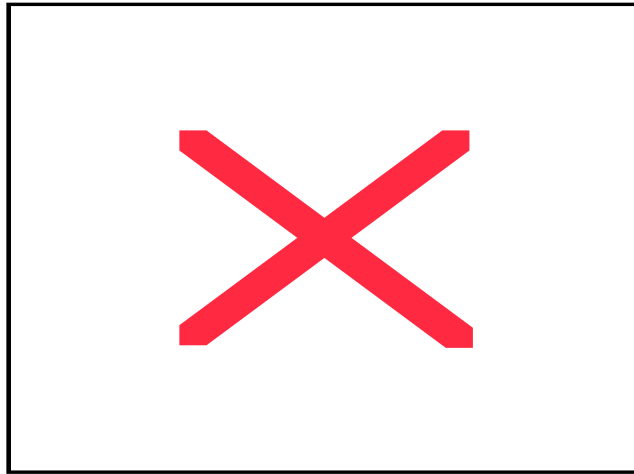
**Figure 5.4:** The maximum decibel variance, across all variable permutations, for each profile at 100 meters.

Even when the changes to the variables are taken in concert (Figure 5.4) the greatest total variance seen at 100 meters was 1.7 dB for the 2-hill profile (Hills), while the rest ranged around 1 dB of difference. This is a very small change for the extremes of the variable value range, and is only twice the error that we witnessed when making the port from the MS Windows version of ITM to the SWAN version. On the whole, such variance could be considered negligible.

Since ITM was originally designed to work with antennas, such as TV broadcast towers that transmit over a very large distance [9], we thought it would be interesting to see whether the environment variables had a greater effect when gain was computed for a greater distance. We therefore expanded the longest single hill profile by an order of magnitude so that instead of portraying a 300m distance, it described a 3 km distance. The effects were slightly less pronounced, and no single computation varied by more than

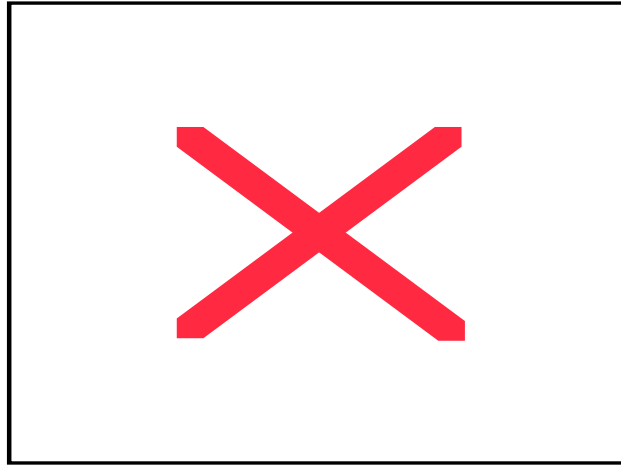
0.3 dB from any other computed signal loss in the group. Even at greater distances, the environmental descriptors do little to change the computed gain.

The most important point to take from the above data is that, even at the extremes of their value ranges, the various parameters for ITM make less difference in the calculated signal attenuation (at most  $\sim 0.2$  dB) than we accepted for differences due to platform and implementation of the port (approximately 0.5 dB). Our results seem to indicate that the model is fairly insensitive to parameter settings. This observation underscores the need for a better understanding of the radio propagation effects modeled by ITM. Although this added investigation is beyond the scope of our study, the practical conclusion we offer is that the user needs not be precise when choosing the values for these environmental variables.



**Figure 5.5:** Path loss for each terrain profile, with identical variable settings. Within each set, from left to right, the distance changes from 50 to 100 to 300 meters.

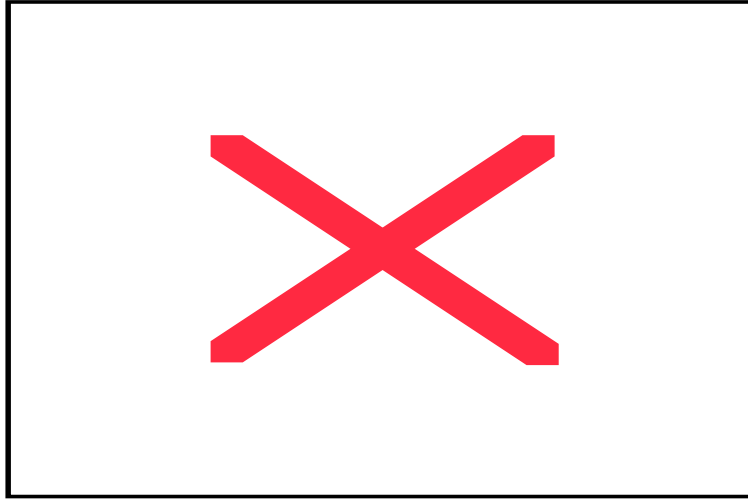
While the various environment parameters are of insignificant importance, both the profile and the distance between transmitter and receiver, as shown in Figure 5.5, play a very significant part in the computed signal attenuation.



**Figure 5.6:** Gain values returned by ITM and FFS/TGR. For ITM, the profile used was the 100m Mid Hill profile. The peak of the hill is at 50 meters.

It is clear that distance and shape of terrain play a role in computed gain and it's interesting to note that ITM's gain computations, shown in Figure 5.6, give exactly the relationship one would expect. Consider an experiment in which we fix the transmitter at one end and move the receiver along the Mid Hill profile, computing gain for each value of Tx-Rx separation. As the receiving antenna moves away from the transmitter the received signal drops off exponentially for the FFS and TGR models. ITM initially models this behavior while the terrain between the two antennas is flat, and even when the receiver moves up the hill, however, as soon as line of sight is lost at approximately 51m, the received signal strength drops precipitously. All of this is in line with the qualitative expectations of signal attenuation on this sort of profile. Clearly, ITM accurately reflects the qualitative signal loss expected from the real world.

It is also useful to note whether ITM and TGR show any significant differences in signal attenuation when using flat terrain, thereby establishing a baseline deviation for the two RF propagation models. As seen in Figure 5.7, there is clear separation between the gain returned by the two models at distances greater than the TGR crossover point at 225m. While it may seem insignificant given that the maximum difference returned by the two models is only 0.39 dB, it must be remembered that since decibels are measured on a logarithmic scale, doubling every 3 decibels, this change is akin to a 10% change in the received signal strength. Although this may seem minimal for any single node, these differences add up to produce higher error when many nodes are simulated (see Section 6).



**Figure 5.7:** The difference in gain between TGR and ITM with a flat profile.

## 6 Wireless Network Simulation with ITM and TGR

After studying ITM and its parameters' sensitivity, we investigated how the RF propagation sub-model affects the simulation of the upper layer protocols.

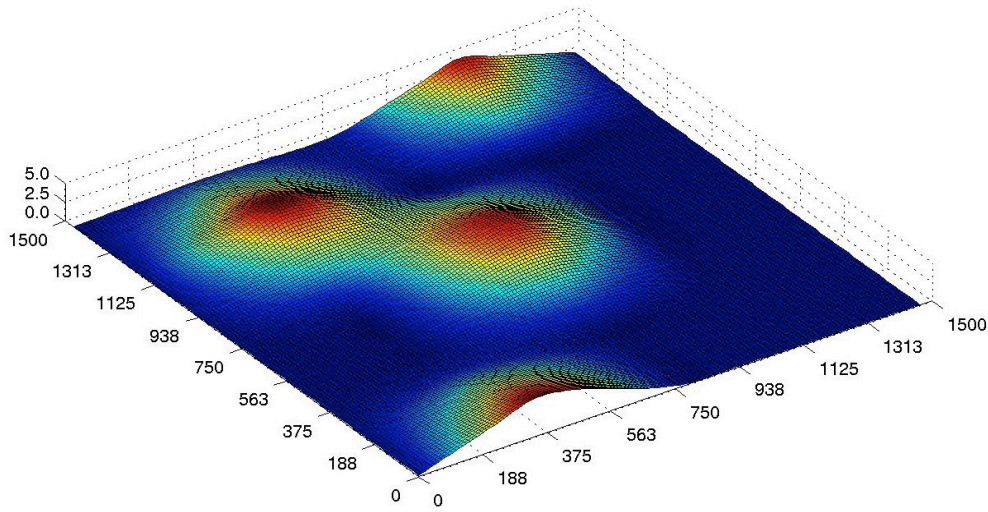
Since our experimentation with ITM's sensitivity to its various parameters showed that, aside from the terrain profile, the variables made little difference at the distances with which we were dealing, we elected to use the lowest values from the ranges listed in Table 5.1. For this round of experiments, we again utilized SOS in order to speed our experimentation. Within our set of experiments, we used 5 varying quantities in order to get a full understanding of the simulated network's performance. Their names, values and descriptions are listed in Figure 6.1.

Name	Values	Description
Terrain	N/A	For ITM, we used a slightly hilly terrain, along with a flat terrain
Transmitting Nodes	20%, 80%	The number of nodes that act as traffic sources
Total Nodes	57, 81	The total number of nodes
Random seeds	N/A	This value seeds the random placement of nodes on the map
Packet Inter-Arrival Time (IAT)	0.25 0.07142857 0.01333333	The time between the end in the transmission of one packet, and the beginning of transmission for the next packet

**Figure 6.1:** The experimental values used in the network simulations



Oftentimes, when one chooses to simulate a terrain with a few small, rolling hills, one will make the assumption that it can be considered virtually flat, and use a model such as TGR. Testing such an assumption was one of the major goals of our research. Accordingly, the terrain that we modeled for our experimentation with ITM was one that contained four, five-meter high hills that rose from 0m to 5m over a distance of approximately 300m. The space is modeled as a square measuring 1.5 km on each side. Figure 6.2 has a rendering of the terrain model that we used.



**Figure 6.2:** The hill terrain used in our simulation. Note that in order to better show the qualitative shape of the landscape, the Z axis is not to the same scale as the X and Y axes.

We also provided ITM with a flat terrain in order to serve as control, and further examine the parity of ITM and TGR. In simulating the network, we used the 802.11b physical and MAC layers, IP, ARP, and AODV routing protocol.

We chose the total numbers of nodes based on a desire for a certain level of connectivity for the wireless network. With an area of  $2.25 \text{ km}^2$ , and an average radius range of approximately 300m, 57 nodes gives an average of 7 neighbors per node. With 81 nodes, we have an average of 10 neighbors per node. We also varied the percentage of sending nodes between 20 and 80 percent. The variance in node density and number of sending nodes provides two additional methods for varying the network traffic load.

The traffic model we used is one that maintains a constant bit rate (CBR). The packet inter-arrival time (IAT) effectively regulates the amount of traffic that any one node is putting onto the network. By using 3 separate values of inter-arrival time, we vary the load placed on the network by each transmitting node between 16, 56 and 300 Kbits/s. By

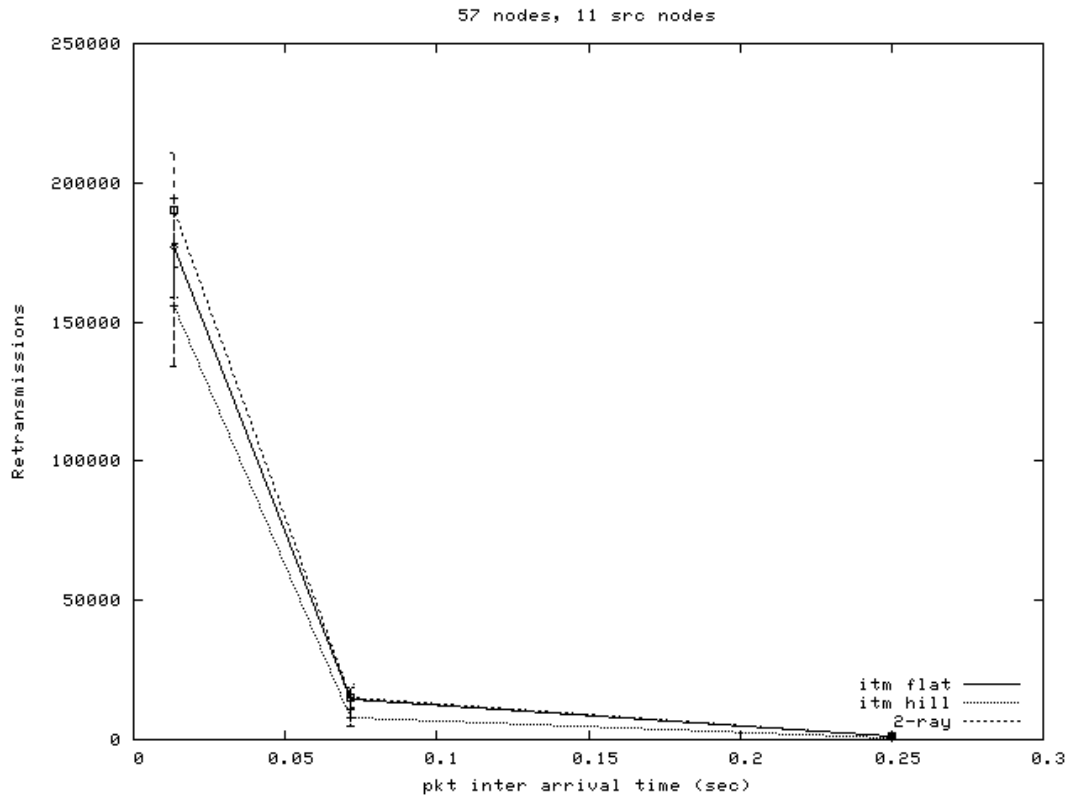
gathering data on end-to-end delay, packet delivery ratio, and number of MAC layer retransmissions, we can observe the effects of a change in the RF propagation model under a variety of network load conditions. This information will help us understand how such changes affect network performance.

Below are the results returned by the 4 experiment groups. The groups are organized in terms of increasing network load, and divided into two sets depending on the fraction of nodes acting as sources of traffic. Each graph is constructed by averaging the values calculated from 10 simulation runs with independent random number streams resulting from different random seeds for each run. The error bars are calculated using 95% confidence intervals.

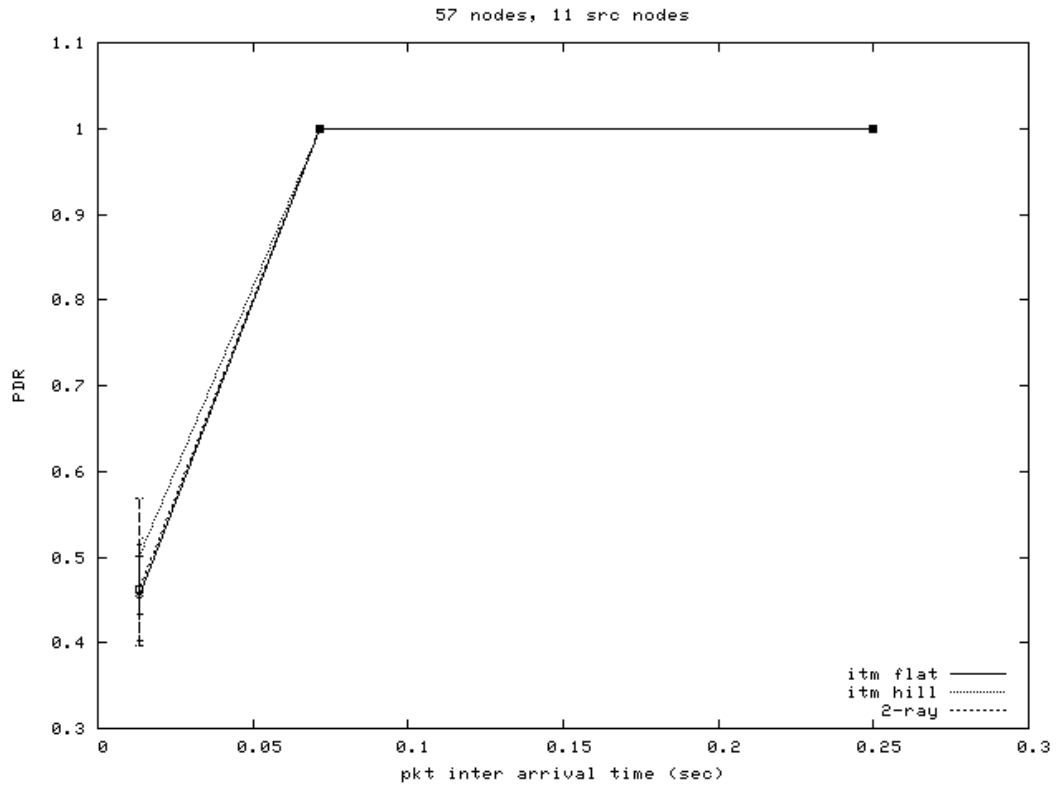
Figures 6.3-6.5 represent the lowest traffic load scenarios in which each node averages 7 neighbors and 20% of the total nodes are transmitters. These curves are generally shaped as one would expect. Packet delivery ratio (PDR) is high while end-to-end delay and packet retransmissions are low for the 16 and 56 Kbps inter-arrival times (IAT). As the IAT shrinks, the network becomes more congested and packet retransmissions and end-to-end delay rise while the PDR drops. This is exactly the general behavior that we would expect from a network undergoing an increase in offered load from a relatively small number of nodes. Network saturation is not reached until the bit rate rises to 300 Kbps.

For the 16 and 56 Kbps IAT values, it is clear that the choice of propagation model makes little difference. The changes in RF propagation sub-models are virtually negligible for both PDR and end-to-end delay, and the confidence interval is very small. At the shortest IAT, ITM Flat and TGR separate slightly from ITM Hill. This behavior is expected given that ITM Flat and TGR simulate the same flat terrain, while ITM Hill uses the rolling hill terrain shown in Figure 2. On the whole, this is exactly the behavior that one would expect when comparing the given RF propagation sub-models since the presence of hills should affect the communication channels of the network in some way.

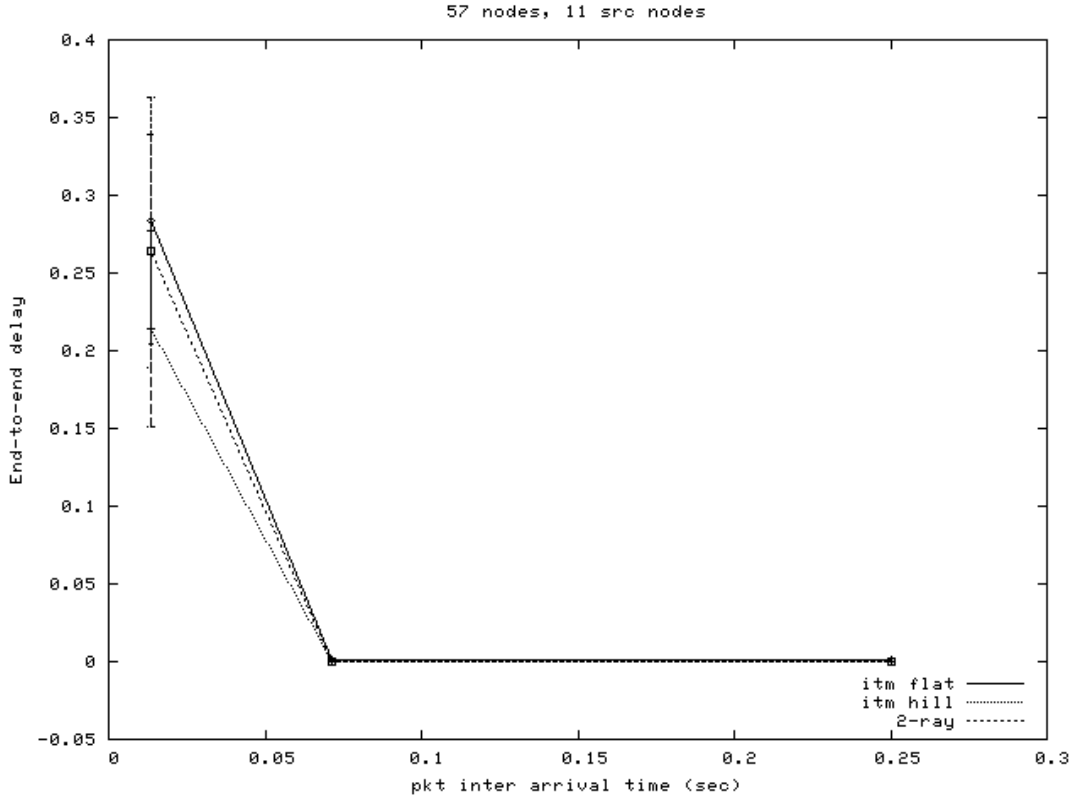
The large increase in the size of the confidence intervals for the 300 Kbps IAT value on Figure 6.5 results from the increase in the number of dropped packets seen in Figures 6.3 and 6.4. As the network load increases, more packets are dropped and so the number of samples used to compute the statistics drops. Since the confidence interval is calculated using the standard formula, the size of the confidence interval will naturally increase as the number of samples drops.



**Figure 6.3:** MAC layer retransmissions for the 57 node, 11 transmitter case.



**Figure 6.4:** Packet delivery ratio for the 57 node, 11 transmitter case.



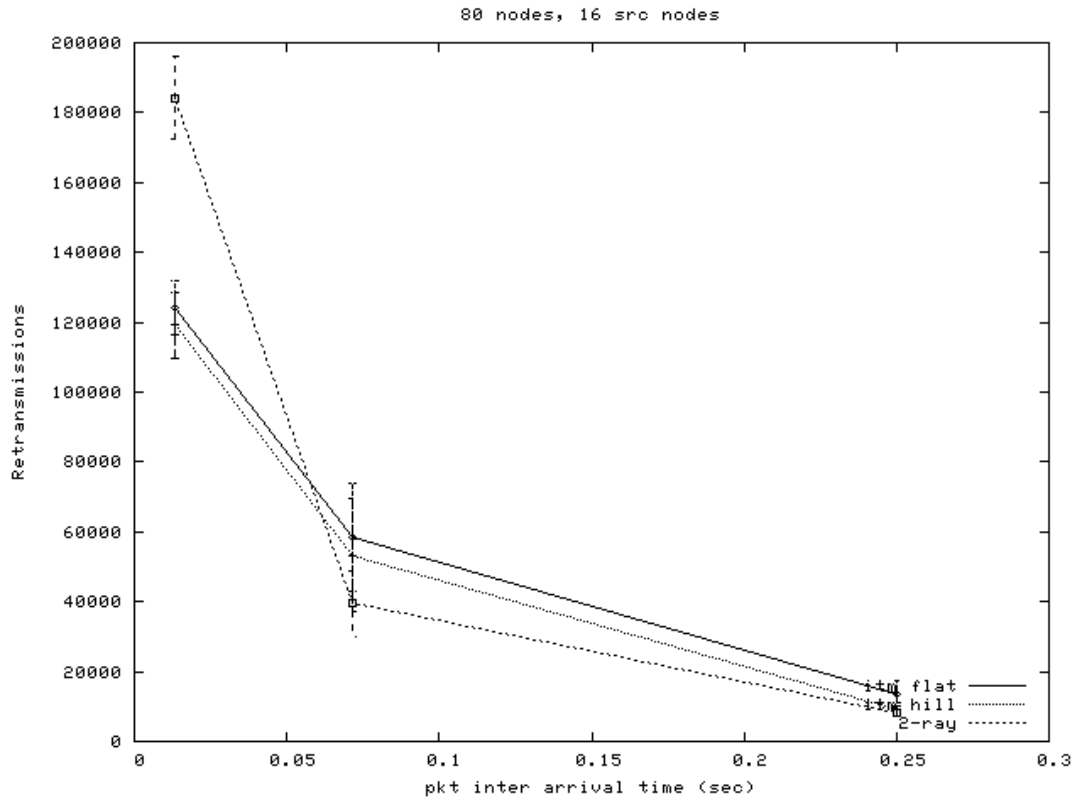
**Figure 6.5:** End-to-end delay for the 57 node, 11 transmitter case.

Figures 6.6-6.8 depict the second lowest traffic scenario with 80 total nodes, 16 of which are transmitting. This results in a higher node density of 10 neighbors per node, with 20% of the nodes offering traffic to the network. The general shape of the curves is similar to those in the first group, and warrants the same conclusion. Traffic load increases in comparison to the first setup because of multiple factors. First, the increased number of transmitting nodes will cause greater load to be placed on the network. Second, the added non-transmitting nodes will send more AODV routing control packets through the network, while the greater average number of neighbors will increase the interference that each node experiences.

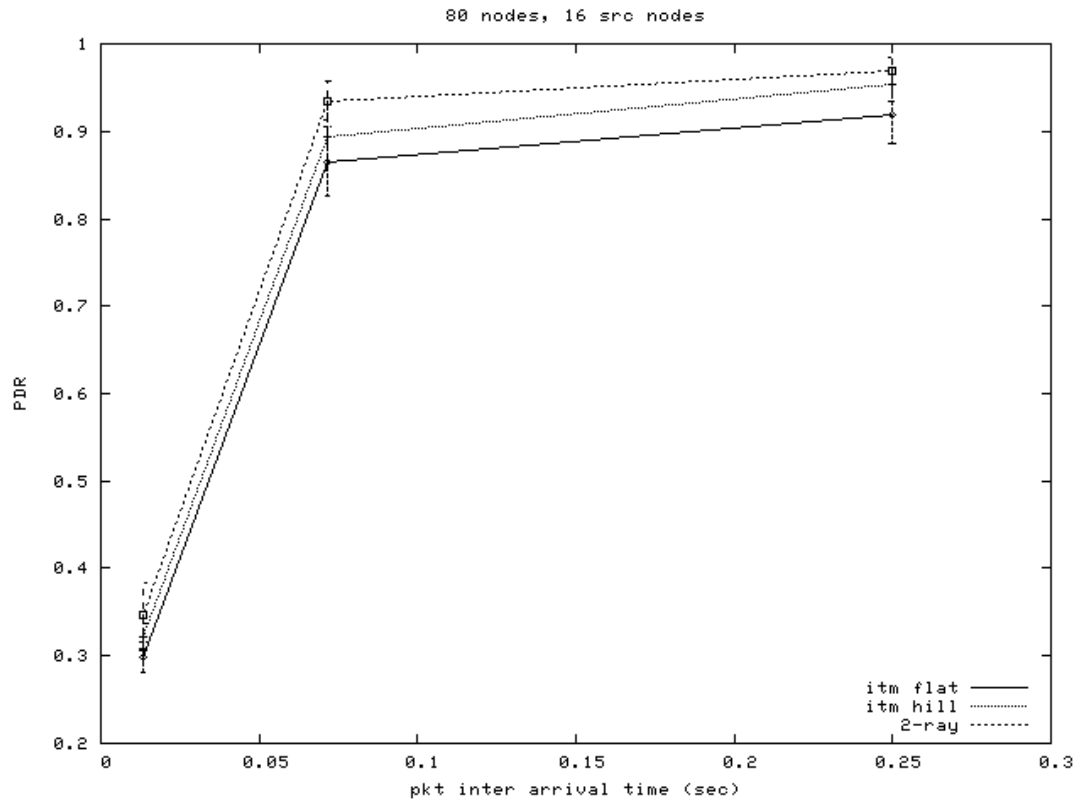
The small increase in the number of transmitting nodes has the largest impact on the performance of the network, and accounts for the growing separation between all of the models for the 56 and 300 Kbps IAT values. An increase in network load leads to increased variance between RF propagation models since even the smallest deviations between the 3 RF propagation sub-models are compounded with every new packet that is placed on the network. For packets whose signal-to-noise ratio was at the limit of the receivable range, small differences in calculated gain may make the received signal too weak, or push the interference level beyond acceptable limits. As more packets are placed on the network, more limit-condition packets will be lost, and the metrics, as we have observed, will be affected.

As is clear from visual inspection, especially in Figure 6.6, the curve for ITM Flat has begun to deviate from that of TGR. This is likely the first evidence of the base-level

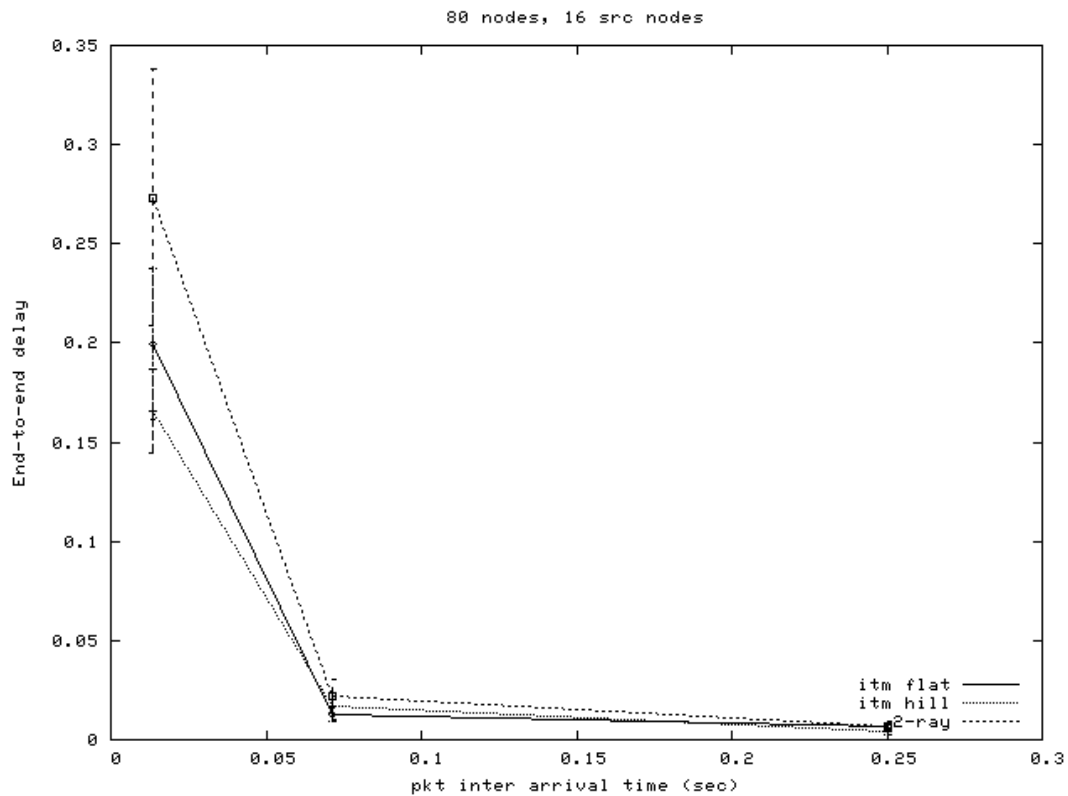
deviation seen in Figure 5.7. Even small deviations like those seen in Figure 5.7 can have a significant effect on the performance of the network because their effects will be compounded for every additional node that is added to the network. Given that the performance of TGR and ITM Flat only continue to deviate in third and fourth groups, we can safely conclude that the base-level deviation has an observable effect on network performance.



**Figure 6.6:** MAC layer retransmissions for the 80 node, 16 transmitter case.



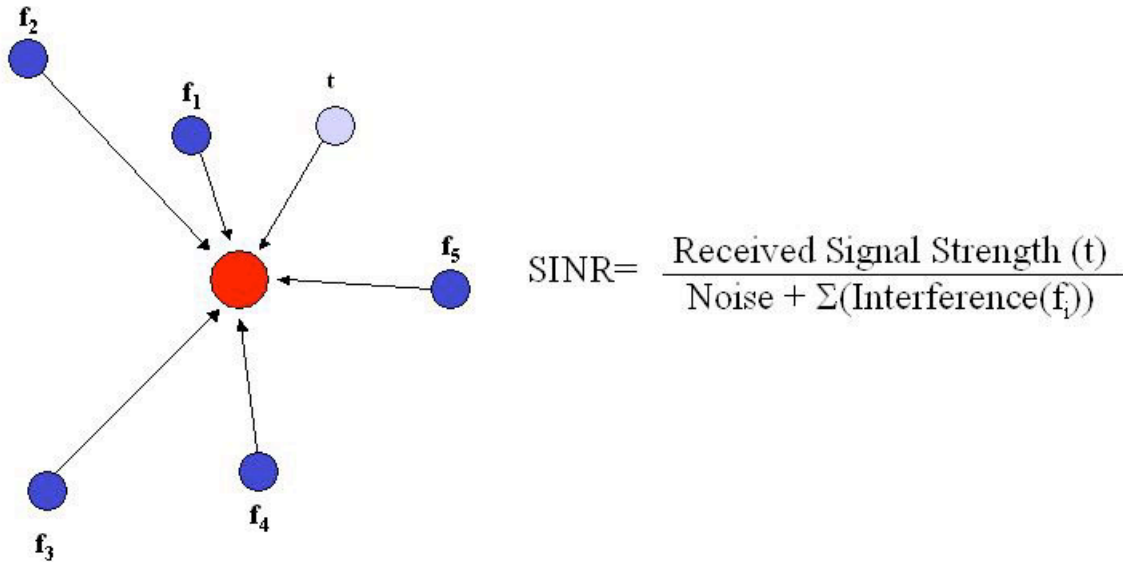
**Figure 6.7:** Packet delivery ratio for the 80 node, 16 transmitter case.



**Figure 6.8:** End-to-end delay for the 80 node, 16 transmitter case.

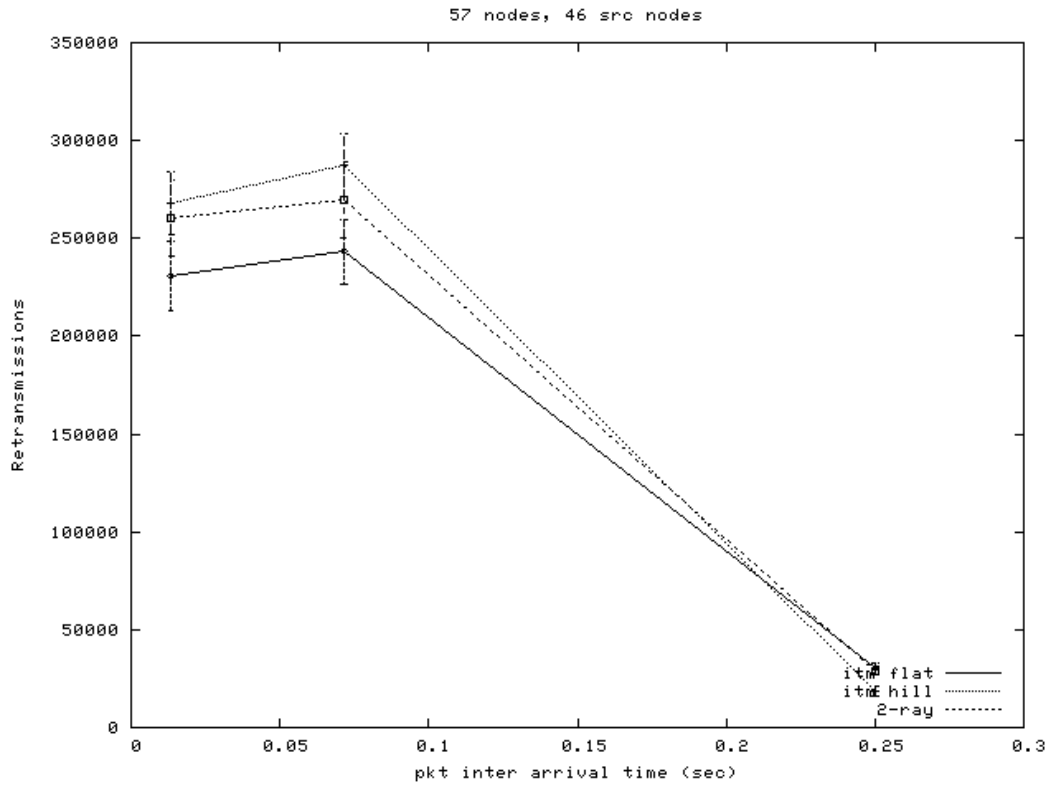
Figures 6.10-6.12 depict the second from highest traffic scenario: 57 total nodes, 46 of which are transmitting. This results in an average of 7 neighbors per node with 80% of the nodes transmitting. The general trends for this set are to be expected. Given the 4-fold increase in the number of transmitting nodes from the first group, we would expect the network to reach saturation more quickly than before, and this indeed occurs. As the IAT decreases, retransmissions quickly climb to a high level and stabilize, while PDR drops to approximately half the level seen in the second set of graphs.

The observed changes in the shape of the curves across groups 3 and 4 are due to a decrease in the signal to interference noise ratio (SINR) whose formula is given in Figure 6.9. Packets are dropped if their calculated SINR is below a given limit, and as the number of transmitting nodes grows, the value of the SINR will obviously decrease. As more packets are dropped, retransmissions will climb, PDR will drop, and end-to-end delay will increase.

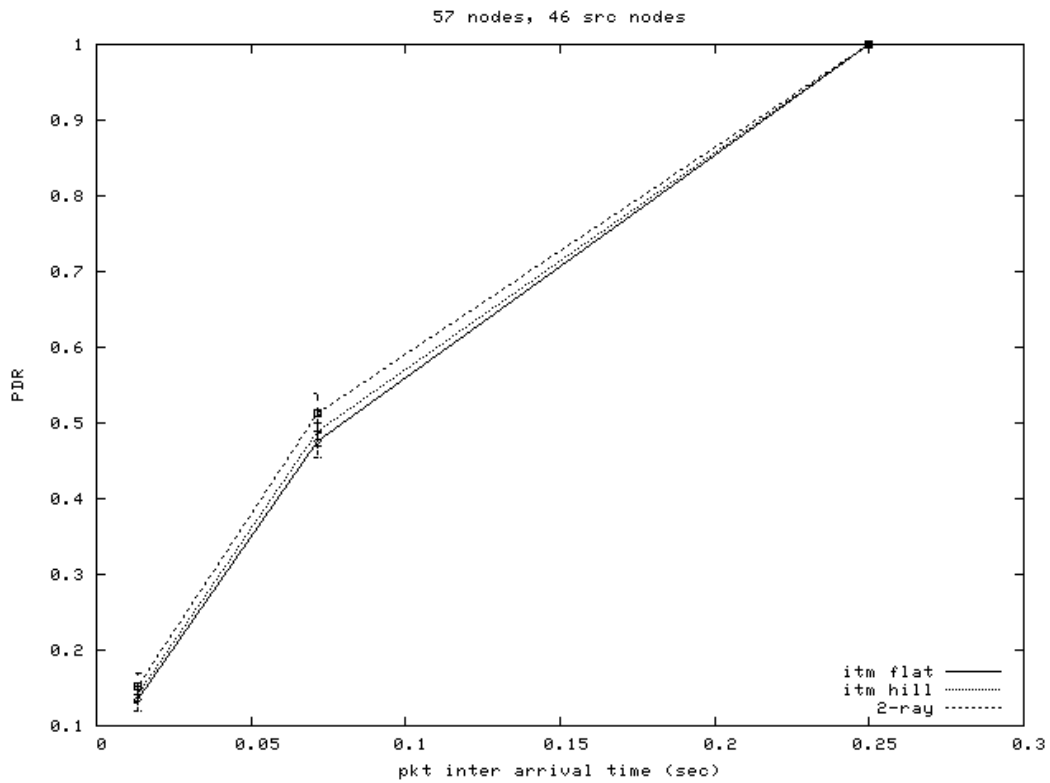


**Figure 6.9:** Definition of SINR.

Figures 6.10-6.12 show increasing deviation between TGR and ITM Flat as the importance of the deviations shown in Figure 5.7 grow with increased network load. Clearly, as network load increases, the choice of RF propagation sub-model becomes much more important to the outcome of the simulation.

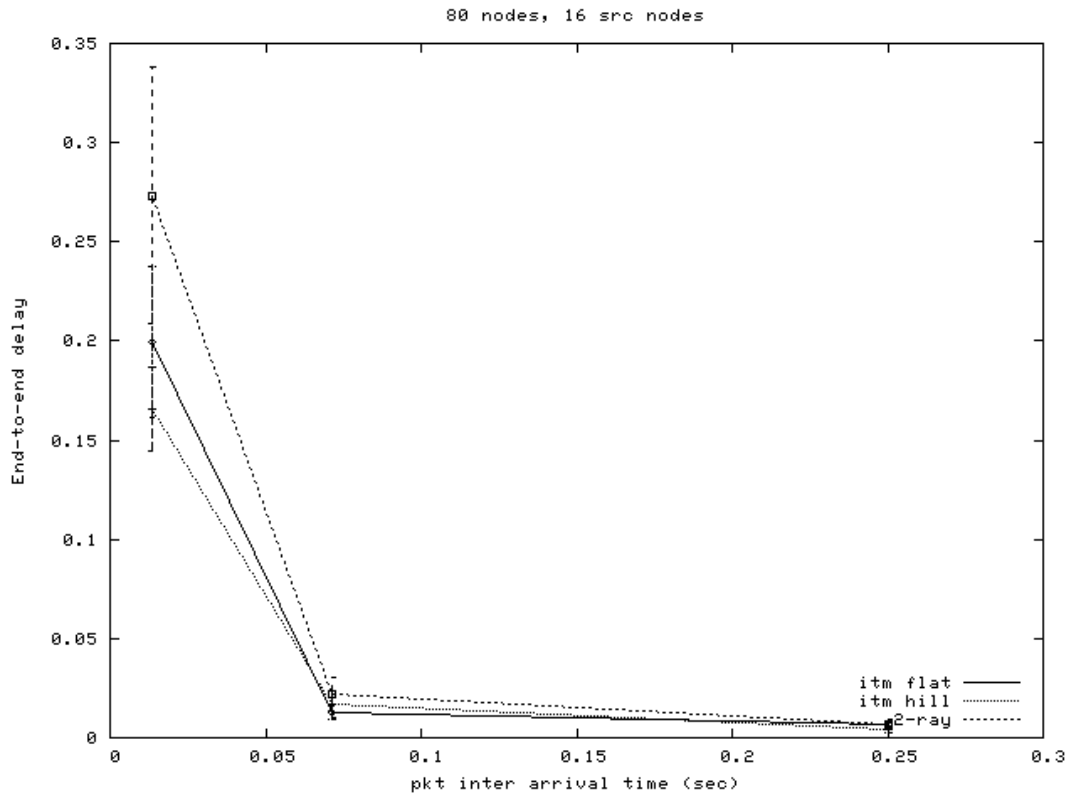


**Figure 6.10:** MAC layer retransmissions for the 57 node, 47 transmitter case.



**Figure 6.11:** Packet delivery ratio for the 57 node, 46 transmitter case.



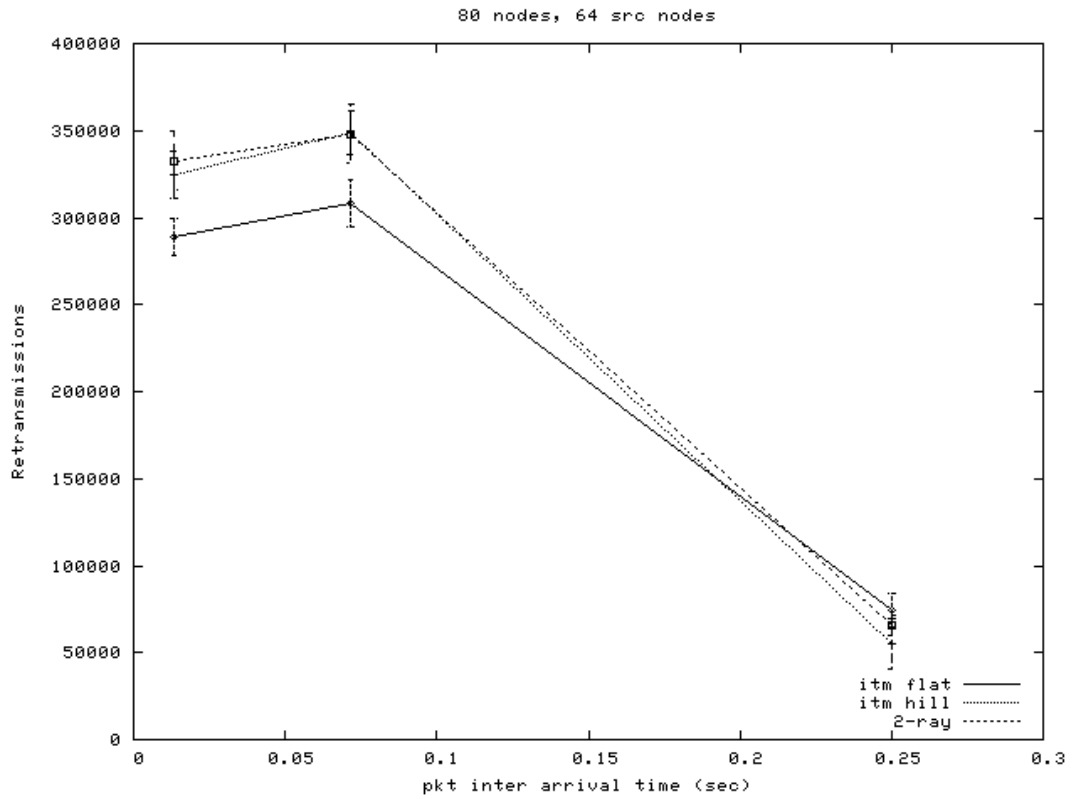


**Figure 6.12:** End-to-end delay for the 80node, 16 transmitter case.

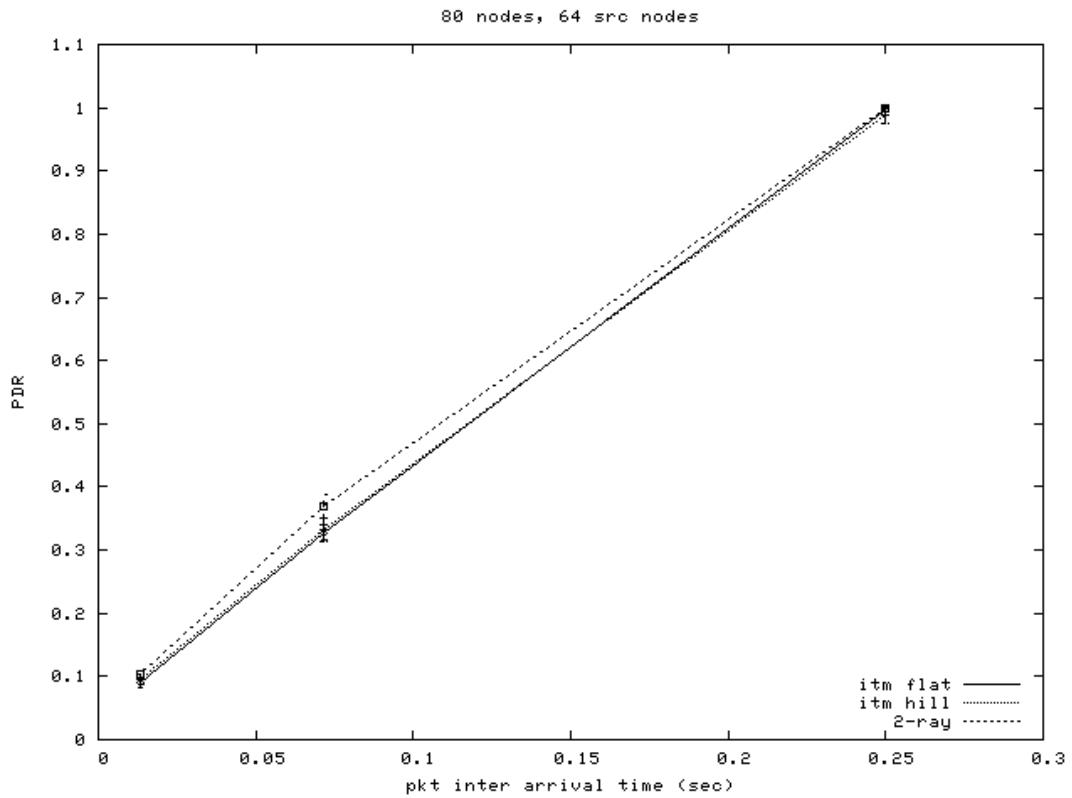
Figures 6.13-6.15 depict the highest traffic load offered to the network. With 80 total nodes giving an average of 10 neighbors and 64 nodes sending traffic through the network, significant continuity with previously observed trends exists. The general conclusion for the shape of these curves is similar to that for the general shape of the curves in the third group.

While the curves in Figures 6.11 and 6.14 may seem oddly similar across all 3 sub-models, this similarity is due to the greatly increased packet loss seen in Figures 6.10 and 6.13. Since packet loss will already be high for the two smallest IAT values, the RF sub-model does not induce much further variability. While the difference due to the choice of sub-model still exists, it has become less visible.

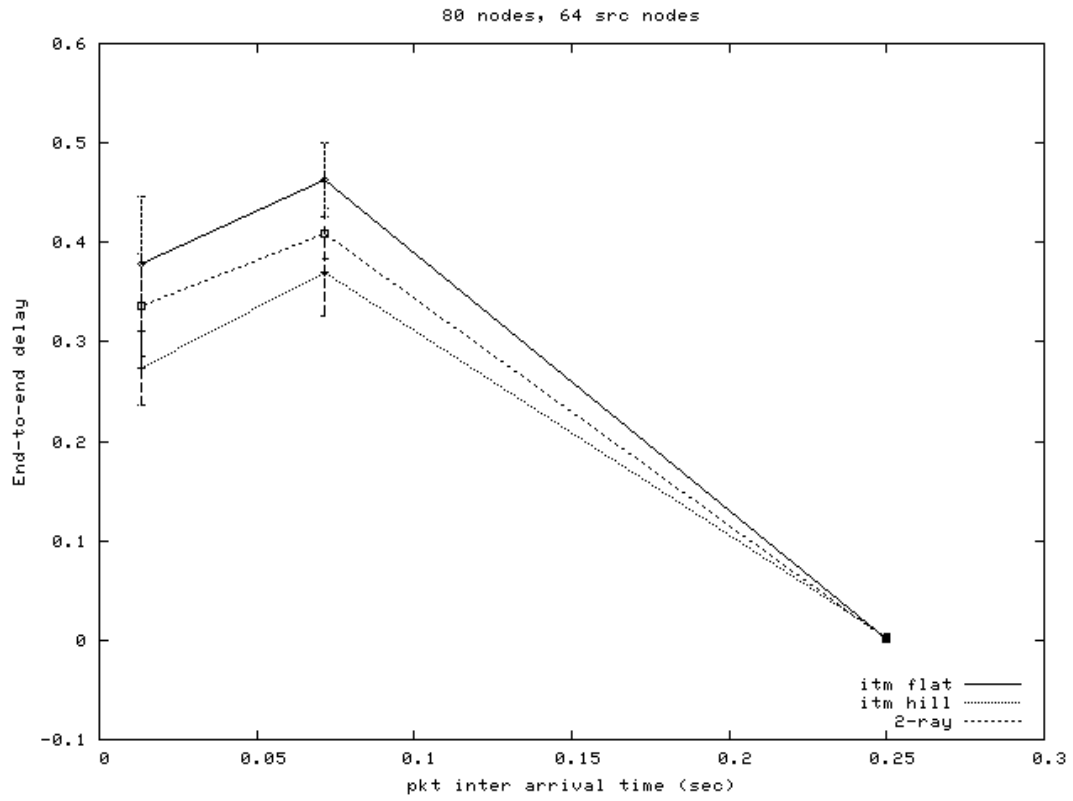
Close examination of all three graphs shows that there is decreasing similarity between ITM Flat and TGR, again showing that the importance of the 0.39 dB base-line deviation shown in Figure 5.7 grows as network traffic increases. With each additional packet that transmitted, the choice of RF propagation sub-model becomes more important.



**Figure 6.13:** MAC layer retransmissions for the 80 node, 64 transmitter case.



**Figure 6.14:** Packet delivery ratio for the 80 node, 46 transmitter case.



**Figure 6.15:** End-to-end delay for the 80 node, 46 transmitter case.

An important feature to note about all of the above graphs is the often-significant size of the confidence interval. In many of the graphs (look to Figure 6.5, for example) the confidence intervals for each of the 3 averaged metric values at a given IAT value overlap to a high degree. While it is out of the scope of this work to do so, the confidence intervals could be narrowed by one of two methods. While computationally intensive, it would be relatively simple to collect more samples for the estimation of each metric. Increasing the number of samples can reduce the width of the confidence interval. If this does not shrink the confidence intervals enough to eliminate overlap, then methods such as Correlated Sampling, defined in [6], could be used to further narrow the confidence intervals. Both these practices would allow us to draw stronger conclusions about the higher layer protocol effects observed above.

## **7 Conclusion**

The implementation of an altitude conscious terrain model literally adds another dimension to the simulation capabilities of SWAN. Along with the terrain sensitive Irregular Terrain Model for simulating radio propagation, terrain sensitive models for node movement and physical processes can now be implemented on an arbitrarily accurate landscape.

Our experience with ITM itself indicates that while ITM has many environment specific variables, their effect on the calculated path loss is limited. This allows the user to have confidence that simulated effects are relatively precise even in incompletely defined environments.

The effects of ITM on the upper protocol layers were examined with a very low, rolling hill terrain model, and the results compared against ITM with a flat terrain and TGR. The results showed some interesting trends suggesting that even the small baseline deviations between ITM and TGR can have observable effects on the upper protocol layer behavior of the simulation. Our findings also suggest that in situations of high traffic load, the choice of RF propagation sub-model can have clear effects on the behavior of the upper layer protocols.

## **8 Future Work**

Along with the development of a terrain model for SWAN and the implementation of ITM within the SWAN architecture, we also designed a set of tools for the creation of an empirical gain map that will accurately define a multidimensional array of path loss values pulled from the real world. We developed a tool that allows the user to divide any plot of land into a grid and assists the user in gathering data to build a digital elevation map for that terrain. The user can thus build a matrix of path loss values that are based in the real world, and use this information to create a new propagation sub-model. This would alleviate any worry of inaccuracies in the calculation of gain, and allow the modeler to be relatively confident that his simulated model represented, at least at the protocol stack's lowest layer, the real world. Future work will involve collecting the data for the gain grid, and implementing a model that uses this information in the computation of signal attenuation in the simulation.

## References

- [1] Bagrodia, Rajive. and Mineo Takai. “MAC protocols in MANETs: Modeling and Simulation”. Tutorial presented at Mobihoc 2002.
- [2] Takai, Mineo, Jay Martin and Rajive Bagrodia. “Effects of Wireless Physical Layer Modeling in Mobile Ad Hoc Networks”. MobiHoc 2001.
- [3] Liu, Jason, L. Felipe Perrone, David M. Nicol, Chip Elliott, and David Pearson. “Simulation Modeling of Large-Scale Ad-Hoc Sensor Networks”. European Simulation Interoperability Workshop 2001.
- [4] DaSSF Project Website,  
<http://www.cs.dartmouth.edu/research/DaSSF>.
- [5] Hufford, G., F. Najmy, A. Paul. ITM codebase and reference manuals,  
<http://elbert.its.blrdoc.gov/itm.html>.
- [6] Banks, Jerry, John S. Carson, II, Barry L. Nelson, and David M. Nicol. *Discrete-Event System Simulation*. Prentice-Hall, 2000.
- [7] L. Felipe Perrone, “SWAN - a Simulator for Wireless ‘Ad Hoc’ Networks”, slides for invited talk, Colorado School of Mines, 12/7/01.
- [8] L. Felipe Perrone, “A Scalable Simulator for TinyOS Applications”, Winter Simulation Conference 2002.
- [9] O. Bendov, “On the validity of the Longley-Rice (50, 90/10) propagation model for HDTV coverage and interference analysis”.  
<http://ww.dielectric.com/broadcast/longley-rice.asp>.
- [10] T. Rappaport, *Wireless Communications Principals and Practice*, second edition. Prentice-Hall, 2001.
- [11] SWAN project website,  
<http://www.cs.dartmouth.edu/research/SWAN/>
- [12] SOS project website,  
<http://www.cs.dartmouth.edu/research/sos/>

## Acknowledgements

I'd like to thank the Institute for Security Technology Studies for giving me both a place to work and the tools to work with. Also Nam Kim, Navam Welihinda and Alison End for listening to my thesis related lamentations. Most importantly, my helpful and understanding advisor Felipe Perrone and the peerless font of all computing knowledge Yougu Yuan . Without their help this could not have been.



## Biography

A long time fan of beige boxes and CRT monitors, **Evan Richardson** has been interested in computer programming since he copied his first program, a random number generator, from a library book sometime during his 5<sup>th</sup> grade year. When not working on his thesis, a rare moment indeed, he enjoys long walks on the beach, people who know how to laugh and love, and making clichéd jokes whenever possible.